**Práctica 5: MLP**

1. Se solicita al usuario los siguientes datos:
   1. El archivo con los datos de entrada (p)
   2. El archivo con los valores deseados (t)
   3. Dado que solo se van a realizar ejemplos de ajuste de señales (MLP->modo regresor).

El usuario deberá indicar el rango en el que se encuentra dicha señal (rango de la señal). Por ejemplo: [-2 2].

* 1. Cada uno de los dataset a aproximar se deberá separar en tres subconjuntos, el usuario podrá elegir una de las siguientes opciones:
     1. 70% train - 15% validation - 15% test
     2. 80% train - 10% validation - 10% test

**Nota:** La separación de los datos debe ser homogénea y cubrir lo mejor posible el rango de señal, para ello se debe programar una función que reciba al vector original de datos y regrese los tres vectores que cumplan estas características.

* 1. El usuario tiene que indicar la arquitectura del MLP se usará para ajustar el polinomio. Por ejemplo:

v1: [1 S1 1] ó [1 S1 S2 1]

v2: [2 1] ó [3 2 1] **Nota:** Los valores en rojo se queda fijos.

**Nota:** Siempre se debe buscar aproximar al polinomio con MLP más pequeño posible. Se usará máximo de dos capas ocultas y no más de 10 neuronas en cada capa oculta.

Se recomienda usar pocas épocas de aprendizaje y observar cómo va el ajuste de la señal. Se debe hacer de manera escalonada y de preferencia no ejecutar un aprendizaje con más de 300 épocas.

Para ello, se debe preguntar al usuario si desea realizar un nuevo aprendizaje o continuar con uno previo. Si se trata de un nuevo aprendizaje se inicializan los parámetros del MLP con valores aleatorios (ver más adelante el paso 2a). Si es continuación en el paso 2a se piden el archivo con los parámetros del MLP, así se podrá hacer muchas épocas de aprendizaje, pero en pasos pequeños. Ejemplo: Deseo hacer 3000 épocas, entonces ejecuto el programa 10 veces y en cada ejecución hago 300 épocas. La primera vez usó valores aleatorios y en las 9 veces restantes, cargo el archivo con los valores finales de la anterior ejecución.

**Nota:** Observe que el archivo de valores finales es diferente al archivo que almacenará los diferentes valores de los parámetros (evolución de pesos y bias) que se generen durante las 3000 épocas, así si el dataset de entrenamiento tuviera 100 valores, las gráficas de evolución de pesos y bias tendrán 300,000 valores.

* 1. El usuario nos da el valor del factor de aprendizaje ().

Se recomienda usarlo en el siguiente rango de 1 x 10-2 a 1 x 10-4 . **Queda prohibido usar valores más grandes o más pequeños.**

* 1. El usuario debe indicar los valores de los criterios de finalización:
     1. Número de épocas máximo: epochmax
     2. Valor del mínimo del aprendizaje: eepoch1x10-4 ó 1x10-5
     3. El valor de cada cuantas épocas durante el aprendizaje se llevará a cabo una época de validación: epochval -> 10% de epochmax.
     4. El número de incrementos consecutivos del error de validación numval. Este valor debe guardar una proporción correcta para que el programa puede activar el Early Stopping.

Ejemplo: epochmax=300, con epochval=10 y numval=3, así le damos hasta 10 oportunidades al Early Stopping de activarse, con numval=5 serían seis oportunidades. En cambio si ponemos numval=30 da 300 épocas por lo que el programa terminaría al mismo tiempo que esta primera validación, en otras palabras cancelamos al Early Stopping y no será posible determinar si hubo sobre entrenamiento.

Estos dos valores se usarán para implementar el algoritmo Early Stopping que es una manera de evitar el sobre aprendizaje (que el error de validación diverge, en vez de converger a cero).

1. Se inicia el aprendizaje del MLP.
   1. Se inicializan todos los valores de pesos y bias de la RNA con valores aleatorios entre -1 y +1
   2. Se inicia la fase del aprendizaje
      1. Dado un valor del contador de época, se verifica si es igual o múltiplo del valor epochval, si es así se realiza una época de validación, si no hace una época de entrenamiento.
      2. **Época de entrenamiento:** Se propaga hacia adelante y se aplican las reglas de aprendizaje a cada uno de los datos del conjunto de entrenamiento. Al final se calcula el error de entrenamiento.
      3. **Época de validación:** Aquí se aplica el algoritmo de Early Stopping

* Se propaga hacia adelante todos los datos del conjunto de validación usando los W y b de la última época de entrenamiento. NO se aplican las reglas de aprendizaje y se calcula el error de validación.
* Se verifica si se cumple numval, si es así se termina prematuramente el aprendizaje, si no se continúa con otra época. Se recomienda iniciar un nuevo experimento, es decir se puede cambiar la arquitectura del MLP y/o el valor del factor de aprendizaje.

1. Se verifica si se cumple algún criterio de finalización como epochmax o que el error de entrenamiento sea menor al dado por el usuario. Si se cumple alguno termina el aprendizaje. Si no se cumple ninguno de los criterios continua el aprendizaje. Una vez finalizado, se pasa al punto.
2. Prueba (generalización del conocimiento): Usando los valores finales del aprendizaje (W y b) se propaga hacia adelante el conjunto de prueba y se calcula el error de prueba.

errortest 1x10-3 a 1x10-4

**Nota importante:**

Se recomienda realizar un aprendizaje por bloques. Esto consiste en poner un número pequeño (epoch=300) de épocas y ver el resultado usando la etapa de despliegue de resultados, si las gráficas indican que el error de aprendizaje está convergiendo a cero, tomen los valores finales de ese bloque y usen con valores iniciales del siguiente bloque de aprendizaje. Sigan usando estos bloques siempre y cuando el error siga bajando a cero y se acerquen a el error de 1x10-4 ó 1x10-5

Durante este procedimiento, deben guardar todos los valores de pesos, bias, error de aprendizaje y error de validación para ser usados en la etapa de despliegue de resultados.

**Etapa de despliegue de resultados:**

Una vez concluida la fase del aprendizaje por alguna de las condiciones de finalización, es decir que no necesariamente se obtuvo un aprendizaje exitoso, se deben realizar lo siguiente:

1. Gráfica de evolución del error de aprendizaje y error de validación.
2. Gráfica de evolución de pesos y bias, realizando una gráfica por cada capa del MLP, por ejemplo si el MLP tiene dos capas, serán dos gráficas separadas (no usar la instrucción subplot).
3. Graficar sólo el conjunto de prueba (usar círculos sin relleno) versus salida del MLP (usar cruz). Se grafican sólo los puntos, no dibujar líneas.
4. Tabla con las distintas arquitecturas de MLP (poner el valor de los vectores V1 y V2) y el valor final de los tres errores: entrenamiento, validación y prueba. En la última fila debe aparecer la arquitectura con la que se obtuvo el mejor resultado.
5. Guardar en el archivo pesos\_y\_bias\_finales.txt, el valor de pesos y bias con el que se obtuvo el mejor resultado.